

## **AUTHENTICATION METHOD FOR ESTABLISHING CONNECTION BETWEEN DEVICES**

### **BACKGROUND OF THE INVENTION**

#### **1. Field of the Invention**

The present invention relates to an authentication method for establishing a connection between devices that can communicate data, and  
5 more particularly, to an authentication method for establishing a connection between devices that want to communicate data in a telecommunications environment using communication specifications such as Bluetooth. The present application is based on Korean Patent Application No. 52658/1999, which is incorporated herein by reference.

#### **10 2. Description of the Related Art**

Bluetooth is a local wireless data communication specification which enables high speed data communications between various electronic devices, using radio frequency (RF) without physical cables. Since Bluetooth adopts Continuous Variable Slope Delta Modulation (CVSD), Bluetooth can send  
15 voice as well as character data without limits in space.

For data communications between devices operating in a telecommunications environment using such communication specifications as Bluetooth, a connection must be established between devices before communicating data. The process for establishing a connection can be broadly  
5 divided into synchronizing radio frequency, establishing a link, and establishing a channel. However, since the specification for such communication specifications as Bluetooth has not been perfectly stipulated yet, research for preparing specifications corresponding to diverse parts, including the connection procedure, are proceeding from various angles.

#### 10 SUMMARY OF THE INVENTION

To solve the above problems, it is an objective of the present invention to provide an authentication method for establishing a connection between devices that want to communicate data in a telecommunications environment using communication specifications such as Bluetooth.

15 To solve the above problems, it is another objective of the present invention to provide an authentication method for performing an authentication procedure to establish a connection between devices that want to communicate data, according to the authentication conditions of the other device with which the present device wants to communicate, in a  
20 telecommunications environment using communication specifications such as Bluetooth.

To solve the above problems, it is an objective of the present invention to provide an authentication method for establishing a connection between devices that can wirelessly communicate data, the method having the steps of (a) sending a first authentication-request message to the other device in order to perform an authentication procedure with the other device to which a connection is wanted; (b) sending a predetermined message according to the current operation mode to the other device and storing the predetermined message when the authentication-response message to the first authentication-request message is received; (c) after performing the step (b), checking whether or not the received first message is a response message corresponding to the predetermined message when a first message from the other device is received; (d) sending a response message corresponding to a second authentication-request message to the other device when the result of checking in the step (c) indicates that the first message is not the response message, but the second authentication-request message; (e) after performing the step (d), checking whether or not a second message is a response message corresponding to the predetermined message when the second message from the other device is received; and (f) finishing the authentication procedure when the result of checking in the step (e) indicates that the second message is a response message corresponding to the predetermined message.

To solve another problem, it is an objective of the present invention to provide an authentication method for establishing a connection between

devices that can wirelessly communicate data, the method having the steps of:

(a) sending a response message corresponding to a first authentication-request message when the first authentication-request message from another device that wants to establish a connection is received; (b) after performing the step

5 (a), checking the authentication condition of the present device when a predetermined message from the other device is received; (c) storing the predetermined message and sending a second authentication-request message to the other device when the result of checking indicates that a mutual authentication is required; and (d) after performing the step (c), sending a

10 response message corresponding to the message stored in the step (c), to the other device when a response message from the other device corresponding to the second authentication-request message is received, and finishing the authentication procedure.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

15 The above objectives and advantages of the present invention will become more apparent by describing in detail a preferred embodiment with reference to the attached drawings in which:

FIG. 1 is a functional block diagram of devices for performing a method according to the present invention;

20 FIG. 2 is a flowchart showing the operation in an authentication-request source in performing an authentication method according to the present invention;

FIG. 3 is a flowcharts showing the operation in an authentication-request destination in performing an authentication method according to the present invention;

FIGS. 4A and 4B illustrate an authentication procedure in a pairing  
5 process; and

FIGS. 5A and 5B illustrate an authentication procedure after a link key is generated.

### **DETAILED DESCRIPTION OF THE INVENTION**

Hereinafter, embodiments of the present invention will be described in  
10 detail with reference to the attached drawings. The present invention is not restricted to the following embodiments, and many variations are possible within the spirit and scope of the present invention. The embodiments of the present invention are provided in order to more completely explain the present invention to anyone skilled in the art.

15 FIG. 1 is a functional block diagram of systems for performing a method according to the present invention. Referring to FIG. 1, the system includes an authentication-request source 100 and an authentication-request destination 110. The authentication-request source 100 and the authentication-request destination 110 are installed in their respective devices that can  
20 communicate data using communication specifications such as Bluetooth. For example, the authentication-request source 100 is installed in a source device, while the authentication-request destination is installed in a destination device.

The authentication-request source 100 and the authentication-request destination 110 both have a host controller interface 102 and 112, a link manager 106 and 116, and an end portion 108 and 118, respectively. Each of the link managers 106 and 116 includes a memory 105 and 115.

5       The host controller interfaces 102 and 112 communicate data and interface, corresponding to layer 2, with their respective hosts (not in drawings). The authentication-request source 100 and the authentication-request destination 110 of FIG. 1 are separated from each host (not in drawings). In a broad sense, the host (not in drawings) is a device operating  
10       according to communication specifications such as Bluetooth, while in a narrow sense, the host is a module installed in a system having multiple functions, and controlled by the central processing unit (not in drawings) of the system so that the module operates in a mode complying with communication specifications such as Bluetooth. This host (not in drawings)  
15       is implemented to perform Logical Link Control and Adaptation Protocol (L2 CAP), which performs a function corresponding to layer 2 in order to establish a channel to each of the host controller interfaces 102 and 112, and the application functions.

      The link managers 106 and 116 are implemented to establish and  
20       release a connection to a device that wants to communicate data according to communication specifications such as Bluetooth, and to perform a function for handling a link established between link managers 106 and 116 of

corresponding devices when a connection is established. Particularly, the memories 105 and 115 store Link Management Protocol (LMP) messages, which are required for establishing a connection between corresponding devices.

- 5           The end portions 108 and 118 perform high frequency processing and baseband control. The high frequency processing is for enabling high frequency communications between devices that want to communicate data in a communication environment complying with Bluetooth, and includes synchronization of high frequency signals and conversion of bits into symbols.
- 10          The baseband control function includes coding/ciphering, packet handling, and frequency hopping.

- Thus, the formed authentication-request source 100 and authentication-request destination 110 synchronize high frequency signals between themselves according to the requests of their respective hosts and then
- 15          establish a connection between themselves. Establishing a connection between themselves is performed through an authentication procedure between the link managers 106 and 116, which are installed in the authentication-request source 100 and the authentication-request destination 110, respectively.

- 20          FIG. 2 is a flowchart showing the operation in the authentication-request source in performing an authentication method according to the present invention, and FIG. 3 is a flowchart showing the operation in the

authentication-request destination in performing an authentication method according to the present invention.

Referring to FIGS. 1 through 3, the authentication method according to the present invention will now be explained.

5 First, the link manager 106 of the authentication-request source 100 sends an authentication-request LMP message (LMP\_au\_rand) through the end portion 108 in step 202, and then the link manager 116 of the authentication-request destination 110 receives the authentication-request LMP message (LMP\_au\_rand) through the end portion 118 in step 302.

10 In step 303, the link manager 116 calculates an authentication-response to the received authentication-request LMP message (LMP\_au\_rand). That is, the authentication-response is calculated using random information, which is included in the received authentication-request LMP message (LMP\_au\_rand), and key information, which the link manager 116 holds. In step 304, the link  
15 manager 116 sends an authentication-response LMP message (LMP\_sres), which is formed from the calculated authentication-response, to the authentication-request source 100 through the end portion 118.

Then, the link manager of the authentication-request source 100 receives the authentication-response LMP message (LMP\_sres) through the  
20 end portion 108 in step 204. In step 206, the link manager 106 checks whether the received authentication-response LMP message (LMP\_sres) is valid. Checking is performed using random information, which is included in the



authentication-request LMP message sent in the step 202, and key information, which the link manager 106 holds. That is, the link manager 106 can compare the result of the calculation, using key information, which the link manager 106 has, and random information, which is included in the authentication-request LMP message (LMP\_au\_rand), with the authentication-response information included in the received authentication-response LMP message (LMP\_sres) so as to determine whether the currently received authentication-response LMP message (LMP\_sres) is valid. At this time, the link manager 106 has the same key information as the link manager 116.

10           When the result of checking in the step 206 indicates that the currently received authentication-response LMP message (LMP\_sres) is not valid, the link manager 106 determines that the corresponding authentication procedure has failed, and a process for failed authentication is performed in step 208. For example, a notice that the corresponding authentication procedure has  
15           been failed can be sent to the corresponding host (not in drawings) and the authentication-request destination 110. Then, the corresponding authentication procedure is finished in step 210.

          However, when the result of checking in the step 206 indicates that the received authentication-response LMP message (LMP\_sres) is valid, the link  
20           manager 106 checks whether the current stage for establishing a connection is a pairing process in step 212. This is determined by whether information on the link key is stored in the memory 105 installed in the link manager 106.

That is, when information on the link key is not stored in the memory 105, it is determined that the current stage for establishing a connection is a pairing process.

When the result of checking in step 212 indicates that the current stage  
5 for establishing a connection is a pairing process, the key used in the step 206 is the initialize-key information. Therefore, a link key which will be used between the link managers 106 and 116 must be generated. The link key is used to authenticate a link to be formed between the link managers 106 and 116 as a result of establishing a connection.

10 Therefore, when the current stage for establishing a connection is a pairing process, the link manager 106 sends an LMP message (LMP\_comb\_key or LMP\_unit\_key) for generating a link key to the authentication-request destination 110 through the end portion 108 and at the same time stores the LMP message in step 214. LMP\_comb\_key is an LMP  
15 message, which is sent when a link key is generated using the result of the combined key information of the link managers 106 and 116, and when there is a message requesting generation of a combination key. LMP\_unit\_key is an LMP message, which is sent when a link key is generated using only key information that the link manager 106 holds, and when there is a message  
20 requesting generation of a unit key.

When an LMP message for generating a link key is sent in the step 214, the link manager 116 checks the authentication condition of the link

manager 116 in step 308. The authentication condition of the link manager 116 is information which enables determination of whether a mutual authentication procedure must be considered. In the present embodiment, Authentication\_Enable information is used for that. For example, when the  
5 Authentication\_Enable information of the link manager 116 is set to '0x00', the link manager 116 determines that the mutual authentication need not be considered. Meanwhile, when the Authentication\_Enable information is set to '0x01', the link manager 116 determines that the mutual authentication needs to be considered.

10           When the result of checking in the step 308 indicates that the mutual authentication need not be considered, then the currently received LMP message is for generating a link, and therefore the link manager 116 sends an LMP message (LMP\_comb\_key or LMP\_unit\_key), which is a response message for generating a link key, to the authentication-request source 100  
15 through the end portion 118 in step 309. At this time, when link key information is generated using the result of the combined key information of the link managers 106 and 116 regardless of the LMP message in the step 214, the LMP message that is sent is a message requesting generation of a combination key (LMP\_comb\_key). When the link key information is  
20 generated using key information, which the link manager 116 holds, the LMP message that is sent is a message requesting generation of a unit key (LMP\_unit\_key).

Then, the link manager 116 generates a link key in step 310. When both LMP messages that are sent and received in order to establish a link key between the link managers 106 and 116 are messages requesting generation of a combination key (LMP\_comb\_key), the link manager 116 generates a link  
5 key based on the combined result. However, when the link manager 106 sends a message requesting generation of a combination key (LMP\_comb\_key) but the link manager 116 sends a message requesting generation of a unit key (LMP\_unit\_key), the link manager 116 generates a link key based on key information of the link manager 116. When both LMP  
10 messages that are sent and received between the link managers 106 and 116 are messages requesting generation of a unit key (LMP\_unit\_key), the link manager 116 generates a link key based on the key information of the link manager 106. When a link key is generated, the authentication procedure in the pairing process is finished in step 314.

15 However, when the result of checking in the step 308 indicates that the mutual authentication needs to be considered, the link manager 116 stores the LMP message, which was received for generating a link key, in the memory 115 in step 316. Then, in step 318, the link manager 116 sends an authentication-request LMP message (LMP\_au\_rand) to the authentication-  
20 request source 100 through the end portion 118.

Meanwhile, after the link manager 106 sends an LMP message for generating a link key in the step 214, when an LMP message from the other

device, the authentication-request destination 110, is received in step 216, the link manager 106 checks the received LMP message in step 218. That is, the link manager 106 determines whether the received LMP message is a message for generating a link key (LMP\_comb\_key or LMP\_unit\_key) or an authentication-request LMP message (LMP\_au\_rand). Checking is performed using identification information (op code) loaded in the payload of the received message. That is, with the identification information, the link manager determines whether the currently received LMP message is an LMP message for generating a link key or an authentication-request LMP message.

10        When the result of checking in the step 218 indicates that the currently received LMP message is an LMP message for generating a link key (LMP\_comb\_key or LMP\_unit\_key), the link manager 106 generates a corresponding link key in step 220. At this time, when the LMP message, which was sent and received between the link managers 106 and 116 to  
15        generate a link key, is a message requesting generation of a combination key (LMP\_comb\_key), the link manager 106 generates a link key based on the combined result. However, when the link manager 106 has sent a message requesting generation of a combination key (LMP\_comb\_key) but the link manager 116 has sent a message requesting generation of a unit key  
20        (LMP\_unit\_key), the link manager 106 generates a link key based on the key information of the link manager 116. When the LMP message that was sent and received between the link managers 106 and 116 is a message requesting

generation of a unit key (LMP\_unit\_key), the link manager 106 generates a link key based on the key information of the link manager 106. Thus, when a link key is thus generated, the authentication procedure of the pairing process is finished in step 210.

5           However, when the result of checking in the step 218 indicates that the received LMP message is an authentication-request LMP message (LMP\_au\_rand), the link manager 106 sends an authentication-response LMP message to the authentication-request destination 110 through the end portion 108 in step 222.

10           Therefore, the link manager 116 of the authentication-request destination 110 receives the authentication-response LMP message (LMP\_sres) in step 320. Then, the link manager 116 checks whether the received authentication-response message is valid in step 322. Checking is performed in the same way as in the step 206. When the result of checking in  
15           the step 322 indicates that the received authentication-response message is not valid, the link manager 116 processes an authentication failure in step 324, and finishes the procedure in step 314. Processing an authentication failure is performed in the same way as in the step 208.

          However, when the result of checking in the step 322 determines that  
20           the received authentication-response LMP message is valid, the next step is generating a link key. The link manager 116 sends an LMP message responding to the LMP message for generating a link key stored in the

memory 115 in step 325. Then, the link manager 116 generates a link key in step 310 in the same way as in the step 326, and finishes the mutual authentication procedure of the pairing process in step 314.

5 The link manager 106 of the authentication-request source receives an LMP message that is a response message for generating a link key in step 224, and then, in step 225, checks whether the received LMP message is a response message corresponding to the message stored in the step 214. When the result of checking indicates that the received message is the corresponding response message, the link manager 106 generates a link key in step 226 in the same way as in the step 220, and then finishes the mutual authentication procedure of the pairing process in step 210. However, when the result of checking in the step 225 indicates that the received message is not the corresponding response message, the link manager 106 awaits the reception of the corresponding response message.

15 Meanwhile, when the result of the checking by the link manager 106 in the step 212 indicates that the current stage for establishing a connection is not a pairing process, the link manager 106 sends an LMP message of connection-establishment-completion (LMP\_setup\_complete) to the authentication-request destination 110 and stores the LMP message in the memory 105 in step 228.

Therefore, the link manager 116 receives the LMP message of connection-establishment-completion (LMP\_setup\_complete) in step 306, and

checks the authentication condition of the link manager 116 in step 308. When the result of checking indicates that the mutual authentication is not needed, the link manager 116 sends an LMP message of connection-establishment-completion (LMP\_setup\_complete), which is a response  
5 message to the received LMP message, to the authentication-request source 100 in step 311. Then, the link manager 116 finishes the authentication procedure and establishes a corresponding connection in step 312, and finishes the procedure for establishing a connection in step 314.

However, when the result of checking in the step 308 indicates that the  
10 mutual authentication is needed, the link manager 116 stores the received LMP message of connection-establishment-completion (LMP\_setup\_complete), received in the step 316, in the memory 115 in step 316. The link manager 116 sends an authentication-request LMP message (LMP\_au\_rand) to the authentication-request source 100 in step 318.

15 The link manager 106 receives the LMP message from the other device, the authentication-request destination 110, in step 230, and checks whether the received message is an authentication-request LMP message (LMP\_au\_rand) or an LMP message of connection-establishment-completion (LMP\_setup\_complete) in step 232. Checking is performed in the same way  
20 as in the step 218.

When the result of checking in the step 232 indicates that the received message is an LMP message of connection-establishment-completion, the link



manager 106 finishes the authentication procedure in step 235, and establishes a corresponding connection, and then finishes the procedure for establishing a connection in step 210. However, when the result of checking in the step 232 indicates that the received message is an authentication-request LMP message  
5 (LMP\_au\_rand), the link manager 106 sends an authentication-response message to the authentication-request destination 110 in step 236.

The link manager 116 receives the authentication-response LMP message in step 320, and checks, as described above, whether the received authentication-response LMP message is valid, in step 322. When the result  
10 of checking in the step 322 indicates that the received authentication-response LMP message is not valid, the link manager 116 performs the step 324. However, if the received authentication-response LMP message is valid, the link manager 116 sends an LMP message of connection-establishment-completion (LMP\_setup\_complete), which is a response message  
15 corresponding to the message of connection-establishment-completion stored in the memory 115, to the authentication-request source 100 in step 327. Then, the link manager 116 finishes the mutual authentication procedure in step 328, and establishes a corresponding connection and finishes the procedure for establishing a connection in step 314.

20 The link manager 106 receives the LMP message of connection-establishment-completion in step 238, and checks whether the received LMP message is the response message corresponding to the LMP stored in the

memory in the step 228, in step 239. When the result of checking indicates that the received LMP message is the corresponding response message, the link manager 106 finishes the mutual authentication procedure in step 240, and establishes a corresponding connection and finishes the procedure for establishing a connection in step 210.

However, when the result of checking in the step 239 indicates that the received LMP message is not the corresponding response message, the link manager 106 awaits the reception of the corresponding response message.

FIG. 4A is a conceptual diagram of the authentication procedure performed in the pairing process and shows a unilateral authentication procedure. Therefore, as shown in FIG. 4A, Authentication\_Enable for the link manager 106 of the authentication-request source 100 is set to '0x01', while Authentication\_Enable for the link manager 116 of the authentication-request destination 110 is set to '0x00'.

In a state in which Authentication\_Enables are set as described, the link manager 106 of the authentication-request source 100 sends an authentication-request LMP message (LMP\_au\_rand) to the link manager 116 of the authentication-request destination 110. Then, the link manager 116 of the authentication-request destination 110 sends an authentication-response LMP message (LMP\_sres), which is a response message to the authentication-request LMP message, to the link manager 106 of the authentication-request source 100.

Then the link manager 106 of the authentication-request source 100 sends an LMP message (LMP\_comb\_key or LMP\_unit\_key) for generating a link key to the link manager 116 of the authentication-request destination 110 and at the same time stores the LMP message in the memory 105. After the  
5 link manager 116 of the authentication-request destination 110 receives the LMP message (LMP\_comb\_key or LMP\_unit\_key) for generating a link key, the link manager 116 of the authentication-request destination 110 sends an LMP message (LMP\_comb\_key or LMP\_unit\_key) for generating a link key, which is a response message to the received LMP message, to the link  
10 manager 106 of the authentication-request source 100, generates a corresponding link key, and finishes the authentication procedure of the pairing process. When the link manager 106 of the authentication-request source 100 receives the LMP message (LMP\_comb\_key or LMP\_unit\_key) for generating a link key, the link manager 106 of the authentication-request  
15 source 100 generates a corresponding link key, and finishes the authentication procedure of the pairing process.

FIG. 4B is a conceptual diagram of the authentication procedure performed in the pairing process, and shows a mutual authentication procedure. Therefore, as shown in FIG. 4B, Authentication\_Enables for both  
20 the link manager 106 of the authentication-request source 100 and the link manager 116 of the authentication-request destination 110 are set to "0x01".

In a state in which authentication-enables are set as described, the link manager 106 of the authentication-request source 100 sends an authentication-request LMP message (LMP\_au\_rand) to the link manager 116 of the authentication-request destination 110. Then, the link manager 116 of the authentication-request 110 sends an authentication-response LMP message (LMP\_sres), which is a response message to the LMP message, to the link manager 106 of the authentication-request source 100.

Then the link manager 106 of the authentication-request source 100 sends an LMP message (LMP\_comb\_key or LMP\_unit\_key) for generating a link key to the link manager 116 of the authentication-request destination 110 and at the same time stores the LMP message in the memory 105.

Since the Authentication\_Enable of the link manager 116 of the authentication-request destination 110 is set to a value which requires a mutual authentication, the link manager 116 stores an LMP message (LMP\_comb\_key or LMP\_unit\_key) for generating a link key in the memory 115 when the LMP message is received. Then, the link manager 116 sends an authentication-request LMP message (LMP\_au\_rand) to the link manager 106 of the authentication-request source 100.

When the link manager 106 of the authentication-request source 100 receives an authentication-request LMP message (LMP\_au\_rand) from the link manager 116 of the authentication-request destination 110 after the link manager 106 of the authentication-request source 100 sent an LMP message

for generating a link key, the link manager 106 of the authentication-request source 100 sends an authentication-response LMP message (LMP\_sres), which is a response message to the received LMP message, to the link manager 116 of the authentication-request destination 110.

5       After the link manager 116 of the authentication-request destination 110 receives the authentication-response LMP message (LMP\_sres), the link manager 116 sends the LMP message (LMP\_comb\_key or LMP\_unit\_key), which is a response message corresponding to the LMP message for generating a key and has been stored in the memory 115, to the link manager  
10   106 of the authentication-request source 100, and then generates a corresponding link key and finishes the mutual authentication procedure of the pairing process. The link manager 106 of the authentication-request source 100 receives the response LMP message corresponding to the LMP message for generating the link key, generates a corresponding link key and finishes the  
15   mutual authentication procedure of the pairing process.

FIG. 5A is a conceptual diagram of the authentication procedure for establishing a connection after a link key is generated, and shows a unilateral authentication procedure. Therefore, as shown in FIG. 5A, the Authentication\_Enable of the link manager 106 of the authentication-request  
20   source 100 is set to '0x01', and the Authentication\_Enable of the link manager 116 of the authentication-request destination 110 is set to '0x00'.

In a state in which Authentication\_Enables are set as described, the link manager 106 of the authentication-request source 100 sends an authentication-request LMP message (LMP\_au\_rand) to the link manager 116 of the authentication-request destination 110, and then the link manager 116 of the authentication-request destination 110 sends an authentication-response LMP message (LMP\_sres), which is a response message to the received LMP message, to the link manager 106 of the authentication-request source 100.

Then, the link manager 106 of the authentication-request source 100 sends an LMP message of connection-establishment-completion (LMP\_setup\_complete) to the link manager 116 of the authentication-request destination 110, and at the same time stores the LMP message of connection-establishment-completion (LMP\_setup\_complete) in the memory 105. The link manager 116 of the authentication-request destination 110 receives the LMP message of connection-establishment-completion, and then sends an LMP message (LMP\_setup\_complete), responding to the received LMP message, to the link manager 106 of the authentication-request source 100. At the same time the link manager 116 finishes the authentication procedure and establishes a corresponding connection. The link manager 106 of the authentication-request source 100 receives the response LMP message (LMP\_setup\_complete) and then finishes the authentication procedure and establishes a corresponding connection.

FIG. 5B is a conceptual diagram of the authentication procedure for establishing a connection after a link key is generated, and shows a mutual authentication procedure. Therefore, as shown in FIG. 5B, the Authentication\_Enables of both the link manager 106 of the authentication-request source 100 and the link manager 116 of the authentication-request destination 110 are set to '0x01'.

In a state in which Authentication\_Enables are set as described, the link manager 106 of the authentication-request source 100 sends an authentication-request LMP message (LMP\_au\_rand) to the link manager 116 of the authentication-request destination 110, and then the link manager 116 of the authentication-request destination 110 sends an authentication-response LMP message (LMP\_sres), which is a response message to the received LMP message, to the link manager 106 of the authentication-request source 100.

Then, the link manager 106 of the authentication-request source 100 sends an LMP message of connection-establishment-completion (LMP\_setup\_complete) to the link manager 116 of the authentication-request destination 110, and at the same time stores the LMP message of connection-establishment-completion (LMP\_setup\_complete) in the memory 105. Since the Authentication\_Enable of the link manager 116 is set to a value which requires a mutual authentication, the link manager 116 of the authentication-request destination 110 stores an LMP message of connection-establishment-completion (LMP\_setup\_complete) in the memory 115 when the LMP

message is received. Then, the link manager 116 sends an authentication-request LMP message (LMP\_au\_rand) to the link manager 106 of the authentication-request source 100.

When the link manager 106 of the authentication-request source 100  
5 receives an authentication-request LMP message (LMP\_au\_rand) from the link manager 116 of the authentication-request destination 110 after the link manager 106 of the authentication source 100 had sent the LMP message of connection-establishment-completion (LMP\_setup\_complete), the link manager 106 of the authentication-request source 100 sends an authentication-  
10 response LMP message (LMP\_sres), which is a response message to the LMP message, to the link manager 116 of the authentication-request destination 110.

After receiving the authentication-response LMP message (LMP\_sres),  
the link manager 116 of the authentication-request destination 110 sends  
15 an LMP message (LMP\_setup\_complete), which is a response message corresponding to the LMP message of connection-establishment-completion that is stored in the memory 115, to the link manager 106 of the authentication-request source 100, and then finishes the mutual authentication procedure and establishes a connection. The link manager 106 of the  
20 authentication-request source 100 receives the LMP message (LMP\_setup\_complete), which is corresponding to the LMP message of connection-establishment-completion (LMP\_setup\_complete) that is stored in



the memory 105, and then finishes the mutual authentication procedure and establishes a connection.

The authentication method for establishing a connection between the link managers described above can be applied to a structure in which the authentication-request source 100 and the authentication-request destination 110 are integrated in a single host (not in drawings).

As described above, the present invention provides a method for performing an authentication procedure between transmitting and receiving link managers according to the authentication condition of the authentication-request destination when a connection is established between devices operating in a communication environment using communication specifications such as Bluetooth, and, therefore, more reliably and precisely establishes connections in a communication environment that operates based on communication specifications such as Bluetooth.